Introduction to Context Sensitive Scanning with X-KEYSCORE Fingerprints

[redacted]

May 2010

Opening Question:

How do you find your target's activity in DNI traffic?

Opening Question:

What if you don't know your targets E-mail address? Or you're trying to find new ones they may be using?

What if the traffic you're interested in doesn't even contain an E-mail address?

What do you do then?

Opening Question:

You may try to look for keywords or patterns to help find your target.

But how do we scan for keywords in the large volumes of data we see in DNI collection? Won't we get too many false hits?

Context Sensitive Scanning

Context sensitive scanning gives analysts a powerful way to surgically target the traffic you're interested in, by only applying the keywords in the manner in which the analyst intended them to be applied

Context Sensitive Scanning

- For example, think about these scenarios:
  - "I want to look for documents from Iran that mention a banned item"
  - "I want to look for people doing web searches on Jihad from Kabul"
  - "I want to look for people using Mojahedeen Secrets encryption from an IPhone"
  - "I want to look for documents containing this regular expression"
  - "I want to look for E-mails that mention words from various categories of interest to CP"
- How would you go about targeting those in passive DNI?

XKS Fingerprints can help!

- Fingerprints are an extremely flexible way to target DNI traffic without the foreknowledge of a strong selector

- They take advantage of X-KEYSCORE's context sensitive scanning engine that has over 70 unique contexts that can be targeted.

- An XKS Fingerprint is simply a meta-data tag that gets applied to a session when a certain criteria is met

- Think of fingerprints as analyst-defined "attributes" of a session

"There's an App for that!"

- There are currently almost 10,000 AppIDs and Fingerprints in X-KEYSCORE – the full list is available from the NSA XKS Home Page

- Odds are there may already be a fingerprint for the traffic you're interested in.

- If not you can easily create your own!

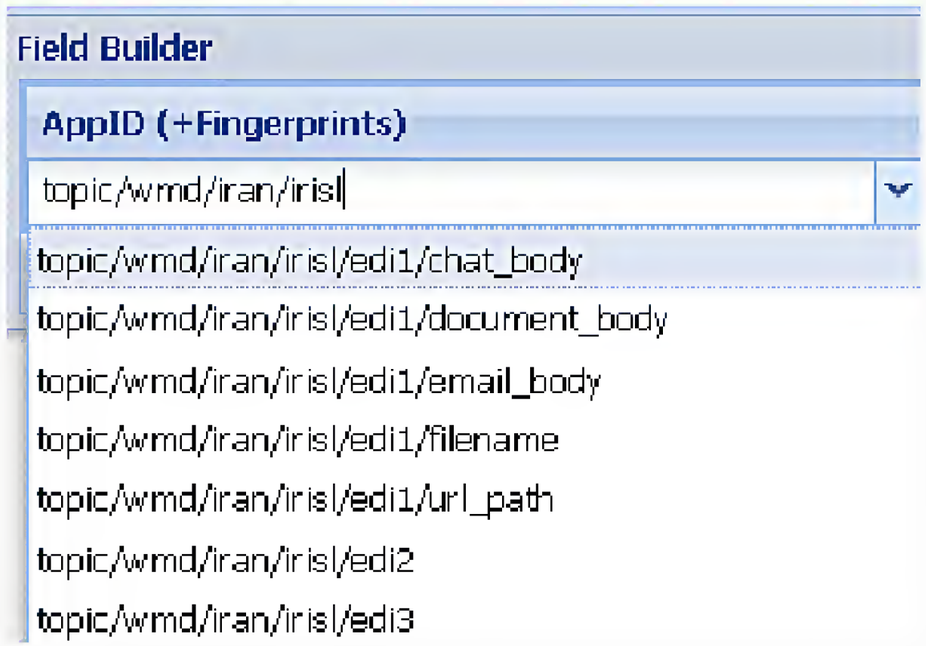
For example

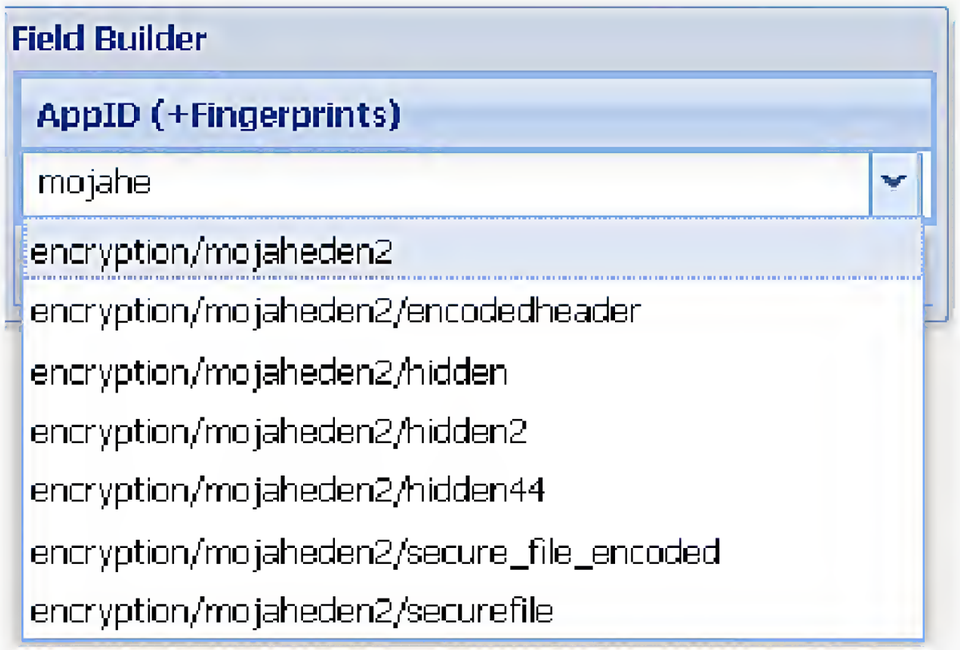
- I'm an analyst in CT – I want to find anytime Mojahadeen Secrets 2 is seen in DNI Traffic.

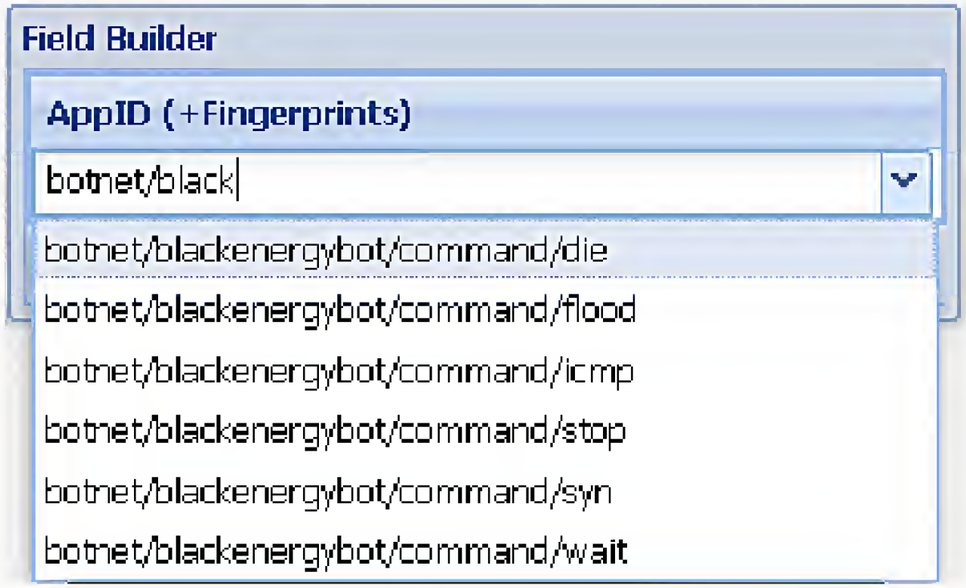
- I'm an analyst in CP – I want to find E-mails or Documents relating to the Iranian Nuclear Procurement network

- I'm an analyst in NDIST/NTOC – I want to find traffic from a known botnet

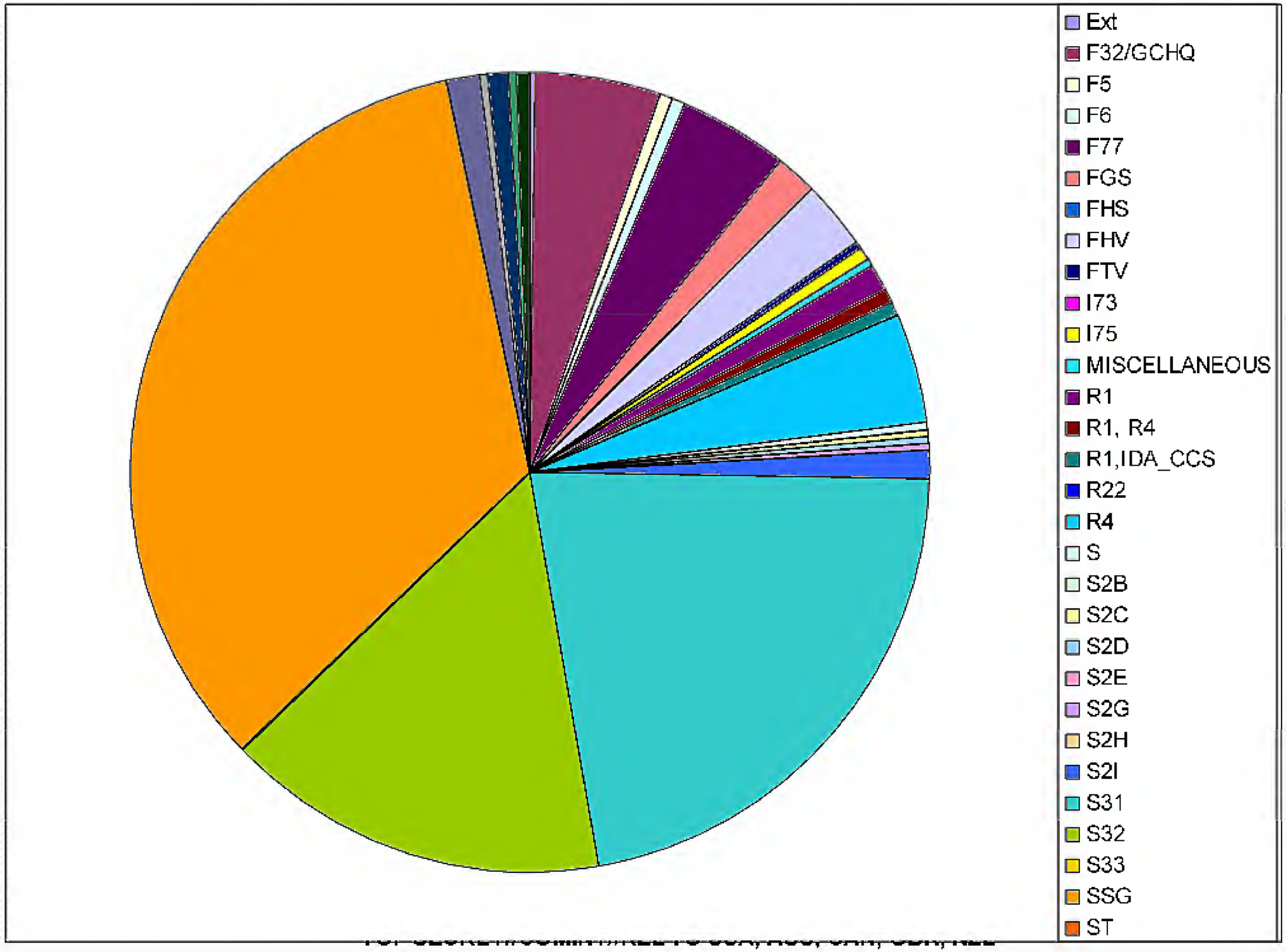
Use Fingerprints!







Who is writing fingerprints?



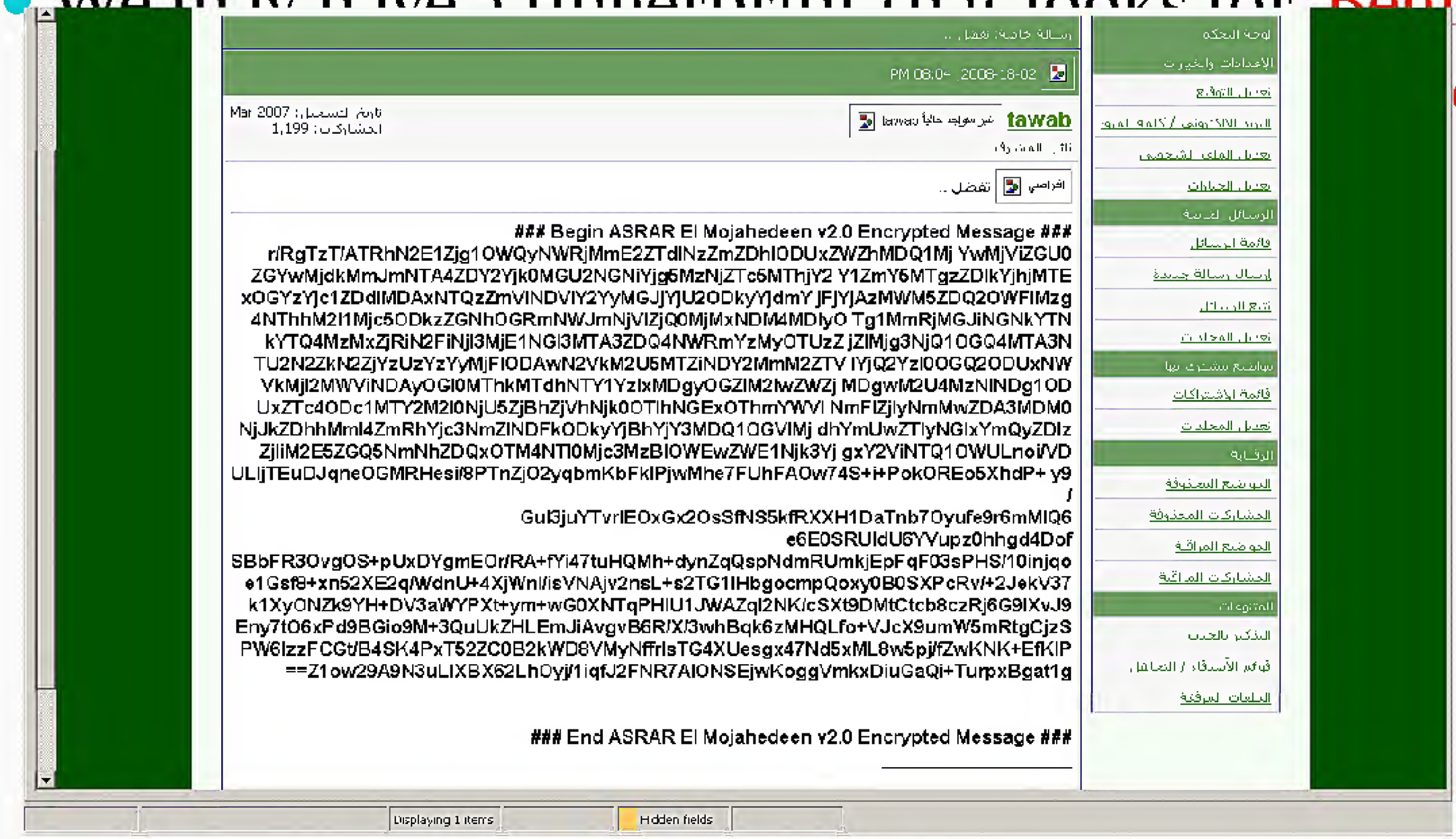
Getting Started

- What are the basics of XKS Fingerprints?

- Simple XKS fingerprints are keyword or regular expression based signatures that are evaluated across the data collected and processed by X-KEYSCORE

Getting Started

- We may have a fingerprint that looks for 'Begin ###' and 'End ###'



Boolean Equations

- Basic fingerprints can also use Boolean equations:

```
appid('voip/sip/IMS', 6.0, wireshark='sip') =
    ('via: sip' or 'v: sip') and 'cseq:' and (
    'p-access-network-info:' or
    'p-called-party-id:' or
    'p-charging-vector:' or
    'p-charging-vector-addresses:' or
    'p-media-authorization:' or
    'security-verify:' or
    'proxy-authorization:' and 'scscf' or
    'path:' and 'pcscf' or
    'path:' and 'scscf'
);
```

Regular Expressions

- And Regular Expressions

fingerprint('encryption/mojahedeen2')=

/(?:Begin|End).ASRAR.El.Mojahedeen.v2\..{0,5}Encrypted.Message/ or

/Mojahedeen.v2\..{0,5}Encrypted.Message/ or

/(?:Begin|End).Al-Ekhlaas.Network.ASRAR.El.Moujahedeen.V2/ or

* Regular expressions must include a fixed "anchor" meeting the minimum keyword length.

Bad: /[A-Z]{3}-[0-9]{3,5}/

OK: /ABC-[0-9]{3,5}/

Binary Patterns

- And Binary Patterns

fingerprint('botnet/IO/XXPW0023') =

$http and

'\x53\x53\x48\x00\x00\x00\x00\x00'c and

'\x00\x00\x00\x00\x03\x00\x53\x4D\x52'c;

fingerprint('botnet/IO/XXPW0023') =

$http and

hex('5353480000000000') and

hex('0000000300534D52');

Positional Logic

```
fingerprint('botnet/IO/XXPW0023') =
    pos('\x53\x53\x48\x00\x00\x00\x00\x00') == 4
    and pos('\x00\x00\x00\x00\x03\x00\x53\x4D\x52') == 24;
```

```
fingerprint('botnet/IO/XXPW0023') =
    $http and
    (pos('\x53\x53\x48\x00\x00\x00\x00\x00') >= 144 and
pos('\x53\x53\x48\x00\x00\x00\x00\x00') <= 184) and
    (pos('\x00\x00\x00\x00\x03\x00\x53\x4D\x52') >= 164 and
pos('\x00\x00\x00\x00\x03\x00\x53\x4D\x52') <= 204);
```

When that's not enough…

- For example, take the first scenario:

"I want to look for documents from Iran that mention a banned item"

- Just using keywords with Boolean equations, how could we restrict the term to only a document body and only coming from Iran?

Context Sensitive Scanning

- X-KEYSCORE's context sensitive scanning engine allows you to explicitly say where you want a term to hit.

- As an early example, the Tech Strings in Documents capability allowed analysts to restrict terms to only Email, Chat or Documents Bodies

- The full XKS Context Sensitive Scanning engine allows for over 70 unique contexts to be used as part of an fingerprint

Context Sensitive Scanning

- For example, take the first scenario:

"I want to look for documents from Iran that mention a banned item"

- Using the XKS context for Country Code (based on NKB information) and the XKS context for Document Bodies, this easily becomes:

fingerprint('demo/scenario1') =
cc('ir') and doc_body('banned item')

Context Sensitive Scanning

- As another example, let's say we want to tag all Iphone usage

- Using the XKS context for User Agent this easily becomes

```
fingerprint('demo/scenario2') =
    user_agent('iphone');
```

USSID18/HRA Considerations

- XKS Fingerprints may not be USSID18 or HRA compliant if they are queried on by themselves

- For example, we may want to fingerprint the use of mobile web devices like the IPhone, so that attribute could be used as part of a more complex query.

- But querying for the IPhone fingerprint itself would be a USSID18 and HRA violation.

USSID18/HRA Considerations

- **But if you want to look for an IPhone user from an Iranian Proxy accessing his Mail.ru account:**

IP Address: 78.[redacted] Either

AppID (+Fingerprints) [fulltext]:

Field Builder

AppID (+Fingerprints)

browser/cellphone/iphone

Add to Field Close

Field Builder

AppID (+Fingerprints)

mail/webmail/mailru

mail/webmail/mailru

mail/webmail/mailru/attachment

mail/webmail/mailru/post

Context Sensitive Scanning

What contexts are available for use in XKS Fingerprints?

HTTP Activity Contexts (1 of 2)

html_title(expr)	The normalized extracted text web page titles html_title('how to' and 'bomb')
http_host(expr)	The "Host:" name given in the http header. http_host('yahoo.com')
http_url(expr)	Every URL from HTTP GET and POST commands. http_url('/mail/inbox?action=delete')
http_url_args(expr)	All arguments given as part of a URL (ie. all text following the '?' in a URL string) http_url('action=delete')
http_referer(expr)	The "Referer:" URL given in the HTTP header http_referer('http://badwebsite/cp?action=show')
http_language(expr)	The normalized two letter iso-6393 language code as inferred from any http and or html header info http_language('fa' or 'de')

HTTP Activity Contexts (2 of 2)

http_cookie(expr)	The "Cookie:" field given in the http header. http_cookie(/PREF=\d\d[a-z]/)
http_server(expr)	The "Server:" type name in the http header. http_server('GWS/2.1' or 'Apache')
http_user_agent(expr)	The "User-Agent:" field given in the http header. http_user_agent(/Mozilla\/[45]/ or 'Chrome')
web_search(expr)	The normalized extracted text from web searches web_search('ricin' or 'plague')
x_forwarded_for(expr)	The X-Forwarded For IP address from the HTTP Header x_forwarded_for('1.2.3.4')

Protocol Contexts 1 of 2

ip(expr)	The source or destination IP address of the session ip('127.0.0.1')
from_ip(expr)	The source IP address of the session from_ip('127.0.0.1')
to_ip(expr)	Every URL from HTTP GET and POST commands. to_ip('127.0.0.1')
ip_subnet(expr)	IP subnet in CIDR notation. ip_subnet('7.211.143.148/24')
port(expr)	The source or destination TCP or UDP port number. port('22')
from_port(expr)	The source TCP or UDP port number. from_port('22')
to_port(expr)	The destination TCP or UDP port number. to_port('22')

Protocol Contexts 1 of 2

cc(expr)	The country (either to OR from) based on IP address cc('ir' or 'pk')
from_cc(expr)	The source country based on IP address from_cc('ir' or 'pk')
to_cc(expr)	The destination country based on IP address to_cc('ir' or 'pk')
protocol(expr)	The textual form of the IP next protocol. protocol('TCP')
next_protocol(expr)	The textual form of the IP next protocol. ip_next_protocol('17')
mac_address(expr)	The MAC address of the target network device. mac_address('00:16:3E:3F:BD:EF')

Communication Based Contexts

email_body(expr)	The UTF-8 normalized text of all email bodies. email_body('how to' and 'build' and ('bomb' or 'weapon'))
chat_body(expr)	The UTF-8 normalized text of all chat bodies. chat_body('how to' and 'build' and ('bomb' or 'weapon'))
document_body(expr)	The UTF-8 normalized text of the Office document. – Office documents include (but are not limited to) Microsoft Office, Open Office, Google Docs and Spreadsheets. document_body('how to' and 'build' and ('bomb' or 'weapon'))
calendar_body(expr)	The UTF-8 normalized text of all calendars. An example is Google Calendar. calendar_body('wedding')
archive_files(expr)	Matches a list of files from within an archive. For example is a ZIP file is transmitted, all names of files within are passed to this context. archive_files('bad.dll' or 'virus.doc')
http_post_body(expr)	The UTF-8 normalized text HTTP url-encoded POSTs. http_post_body('action=send' and 'badguy@yahoo')

Communication Based Contexts Aliases

doc_email_body(expr)	This covers the email_body and document_body contexts doc_email_body('how to' and 'build' and ('bomb' or 'weapon'))
communication_body(expr)	This covers the email_body, document_body and chat_body contexts chat_body('how to' and 'build' and ('bomb' or 'weapon'))

Context sensitivity

Why use context-sensitive scanning?

- More intuitive - you can say what you mean
- More accurate - if 'maps.google.com' is mentioned in a blog post, you don't want to try processing it as a Google Maps session
- Better performance for XKEYSCORE

Examples

- "I want to look for people doing web searches on Jihad from Kabul"

- Using the from_city() and web_search() context this becomes

fingerprint('demo/scenario3') =
    from_city('kabul') and web_search('jihad');

Examples

- "I want to look for people using Mojahedeen Secrets encryption from an IPhone"

- You can even use existing fingerprints in a fingerprint definition! So this becomes:

fingerprint('demo/scenario4') =
fingerprint('encryption/mojahdeen2' and
fingerprint('browser/cellphone/iphone')

Examples

- "I want to look for documents containing this regular expression"
- Using doc_body this becomes:

fingerprint('demo/scenario5') =
doc_body(/blah[a-z]{3-5}something/)

Example 4

- "I want to look for E-mails that mention words from various categories of interest to CP"

- You can use multiple variables in an equation like this:

```
topic('wmd/acw/govtorgs') =
    email_body($acwitems and $acwpositions and
  ($acwcountries or $acwbrokers or $acwports));
```

Example 4

- $acwitems = 'machine gun' or 'grenade' or 'AK 47'
- $acwpositions = 'minister of defence' or 'defense minister'
- $acwcountries = 'somalia' or 'liberia' or 'sudan'
- $acwbrokers = 'south africa' or 'serbia' or 'bulgaria'
- $acwports = 'rangood' or 'albasra' or 'dar es salam'

```
topic('wmd/acw/govtorgs') =
    email_body($acwitems and $acwpositions and
  ($acwcountries or $acwbrokers or $acwports));
```

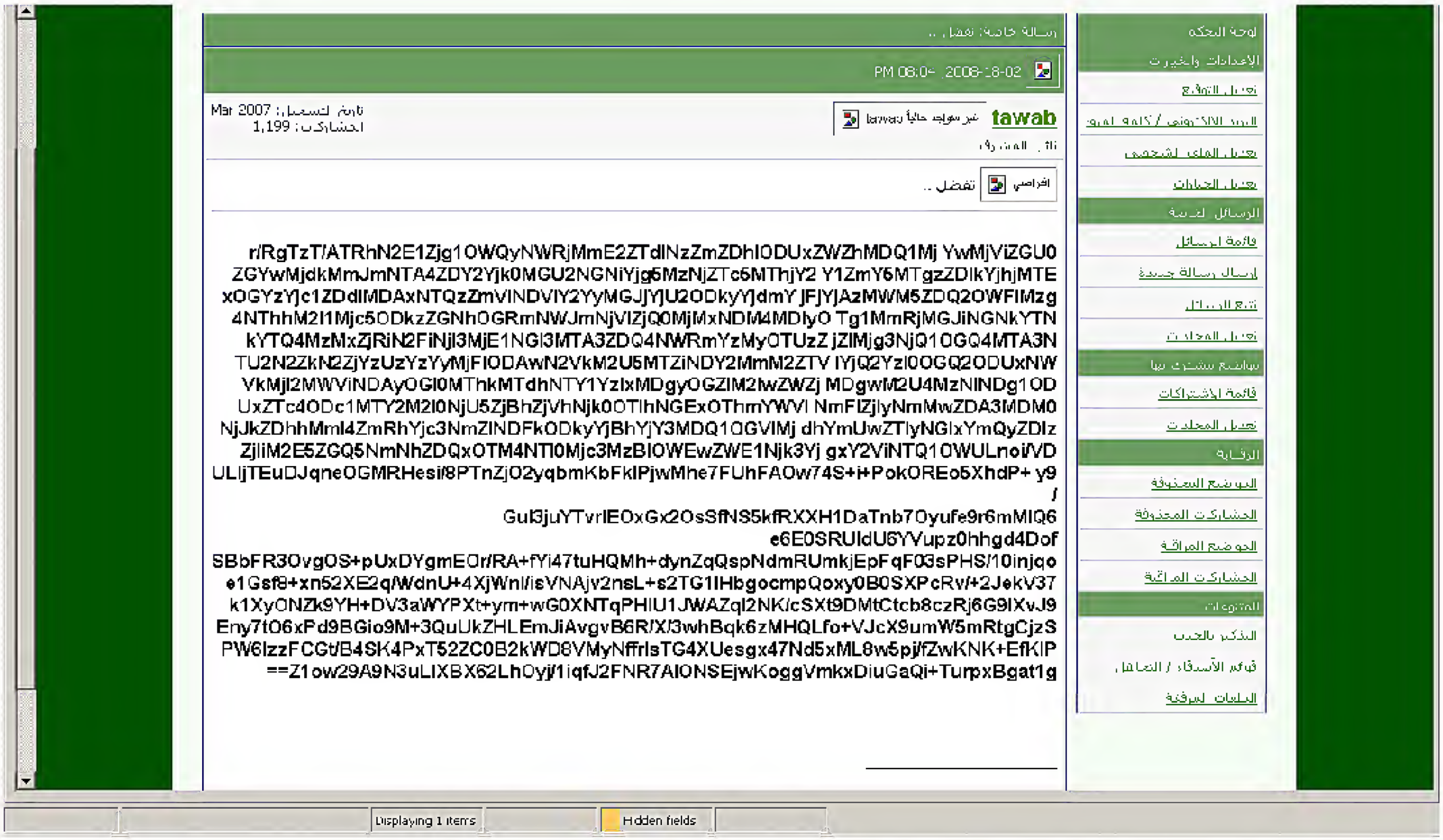
Advanced Code-Based Fingerprints

- What happens when there are no keywords or regular expressions that will help identify the traffic of interest to you?

- As enough example, many of the CT Targets are now smart enough to not leave the Mojahedeen Secrets header in the E-mails they send. How can we detect that the E-mail (which looks like junk) is in fact Mojahedeen Secrets encrypted text

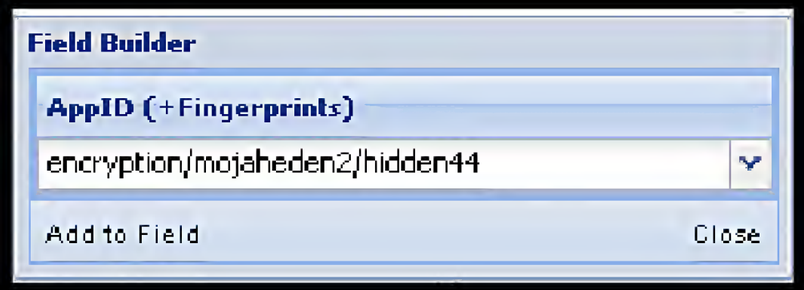
- A C++ code fingerprint can help evaluate that data

Code Based Fingerprint



Advanced Code-Based Fingerprints

```
fingerprint 'encryption/mojaheden2/hidden44' =
        $moj_cipher_first_test
        : c++ extractors : {{
        msg1 = /([a-zA-Z0-9\+\/]{6}[DHLPTXbfjnrvz37\/][4-9\+\/])AT([ABEFIJMNQRUVYZcdghkl][hijklmwxyz012345][MNOYZ][DGTWjnrz2])(12))/g;
        msg2 = /([a-zA-Z0-9\+\/]{6}[DHLPTXbfjnrvz37\/][4-9\+\/])AV([ADEFIJMNQRUVYZcdghkl][hijklmwxyz012345][MNOYZ][DGTWjnrz2])(12))/g;
        }}
        main : {{
          std::string msg;
          if (msg1)
            msg = msg1[0];
          else if (msg2)
            msg = msg2[0];
          else
            return false;

          char buf[16];
          char chunk1[16];
          char chunk2[16];
          char chunk3[16];
          if(true){
            snprintf(chunk1, 16, "%02x%02x%02x%02x",
                msg[10] & 0xff,
                msg[11] & 0xff,
                msg[12] & 0xff,
                msg[13] & 0xff);
            snprintf(chunk2, 16, "%02x%02x%02x%02x",
                msg[14] & 0xff,
                msg[15] & 0xff,
                msg[16] & 0xff,
                msg[17] & 0xff);
            snprintf(chunk3, 16, "%02x%02x%02x%02x",
                msg[18] & 0xff,
                msg[19] & 0xff,
                msg[20] & 0xff,
                msg[21] & 0xff);
            if(!((strcmp(chunk1, chunk2) == 0) ||
               (strcmp(chunk2, chunk3) == 0) ||
               (strcmp(chunk1, chunk3) == 0))){
              std::string msg_decoded = xks::base64dec(msg);
              for(size_t i = 7; i < msg_decoded.size(); i++){
                if(msg_decoded[i] < '0' || (msg_decoded[i] > '9' && msg_decoded[i] < 'a') || msg_decoded[i] > 'f'){
                          return false;
                }
              }
              snprintf(buf, 16, "%02x%02x%02x%02x",
                msg_decoded[1] & 0xff,
                msg_decoded[2] & 0xff,
                msg_decoded[3] & 0xff,
                msg_decoded[4] & 0xff);
              std::string keyid_hex = buf;
```



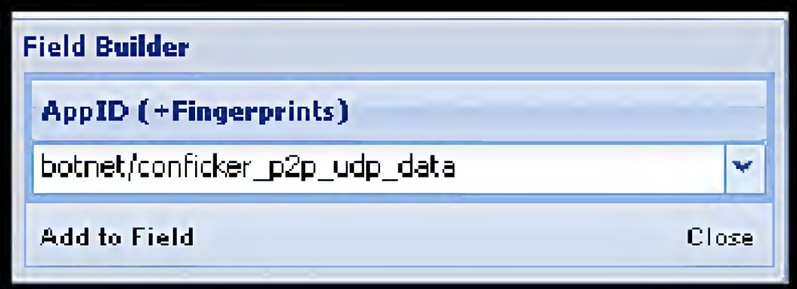
Advanced Code-Based Fingerprints

- As another example, some of the activity from the Conficker botnet simply can't be detected with keywords or regular expressions

- In cases like this, C++ code can be used inside a fingerprint to test the data further

Advanced Code-Based Fingerprints

```
fingerprint('botnet/conficker_p2p_udp_data', 7.3) =
  $udp and not 'to ap 00' and not 'fm ap 00
  : C++{{
  // Classification: CONFIDENTIAL//REL TO USA, FVEY
  // NOT releasable to third-parties
  uint8_t key8;
  uint8_t key9;
  uint8_t pkt_type;
  uint8_t decrypted_bytes[4];
  uint32_t running_hash = 0;
  uint32_t K_high;
  uint32_t K_low;
  uint32_t stored_hashes[4] = {0,0,1,0};
  uint32_t min_pkt_len;
  uint32_t max_pkt_len;
  uint32_t t;
  packet_t pkt;

  while(pkt = get_packet())
  {
  if (pkt.size < 10)
    return false;

  key8 = (uint8_t)(pkt.data[7]<<1 | (pkt.data[7]>>7)&1));
  key9 = (uint8_t)(key8<<2 | ((pkt.data[7]>>5)&1));

  if (((key9 ^ pkt.data[9])&0xfc) != 0x80)
    return false;  // Not Conficker, so abort
  if ((key9 ^ pkt.data[0]) & 0x02)
    return false;  // bit not set for UDP packets

  if (pkt.size<23)
    continue;

  if ((key9 ^ pkt.data[9]) !=0x80)
    continue;

  pkt_type = (key8 ^ pkt.data[8])>>3;

  if(pkt_type & 0x10) // summary
    continue;

  if(!(pkt_type & 0x08)) // not a data packet
    continue;

  min_pkt_len = 23;
  max_pkt_len = (uint32_t)pkt.size;

  K_high = uint32_t(pkt.data[7])<<24u | uint32_t(pkt.data[6])<<16u | uint32_t(pkt.data[5])<<8u | uint32_t(pkt.data[4]);
  K_low  = uint32_t(pkt.data[3])<<24u | uint32_t(pkt.data[2])<<16u | uint32_t(pkt.data[1])<<8u | uint32_t(pkt.data[0]);

  running_hash = 0;

  for(t=8; t<max_pkt_len; t++)
    {
      if(t>=8)  // decrypt data
        {
```



Meta-data Extracting Fingerprints

- What happens when you find data and want some pieces of meta-data extracted?

- XKS Fingerprints can be used to extract meta-data to select XKS database tables.

- Or if no existing database is applicable, you can define your own database schema for the meta-data

Free File Upload Sites

- As a real life example, think of all of various Free File Upload (FFU) sites of interest
- When a user uploads a document they get a response page that looks like this:

Free File Upload Sites

Welcome to ZSHARE

With zSHARE you can upload files, images, videos, audio and flash for free. Simply use the upload form below and start sharing! You can also use zSHARE as your personal file storage: backup your data and protect your files. First Time? Read our FAQ!

- Upload now
- Login
- Create Free Account
- Premium
- FAQ

File Uploaded

The file **khi pics.zip** was successfully uploaded! (4.04MB). You're now ready to share it with unlimited people or keep it as a backup.

Download Link

http://www.zshare.net/download/637199570b174c9f/

Link for forums: [URL=http://www.zshare.net/download/637199570b174

Direct Link: http://www.zshare.net/download/637199570b174c9f/

Delete Link: http://www.zshare.net/delete.html?63719957-7c8893b1

E-mail Me This Info

To receive all the info on the file you uploaded, such as **removal instructions** and **download link**, enter your e-mail address on the field below:

Your e-mail:

Free File Upload Sites

- Look at all the great information on that page:

File Uploaded

The file **khi pics.zip** was successfully uploaded! (4.04MB). You're now ready to share it with unlimited people or keep it as a backup.

Download Link

http://www.zshare.net/download/637199570b174c9f/

Link for forums: [URL=http://www.zshare.net/download/637199570b174

Direct Link: http://www.zshare.net/download/637199570b174c9f/

Delete Link: http://www.zshare.net/delete.html?63719957-7c8893b1b

Free File Upload Sites

- How can we quickly get that information extracted as Meta-data and be agile enough to respond to each FFU site which may have its own format

- XKS "v4" Fingerprints allow you to use the XKS Fingerprint Language to extract meta-data into the XKS database

- Fingerprints are deployed within an hour of being accepted meaning you no longer need to wait for all 130+ XKS sites to be upgraded to have the latest and greatest capabilities.

Free File Upload Sites

```
appid('filetransfer/web/zshare_net/upload/response', 5.0)=
        http_title('zSHARE') and 'zshare.net/delete.html'
        : c++
  extractors : ({
    wft_file_name = /The\sfile\s<strong><font\scolor=\"#333333\">([^<]{1,300})\s</;
    wft_delete_url = /zshare.net\/delete.html\?([0-9]+)-([0-9a-zA-Z]{32})\"/;
    wft_upload_id =  /<font color=\"#666666\"><a href=\"http:\/\/www\.zshare\.net\/[^\/]+\/([0-9]+)[0-9a-f]{8}/;
    wft_url = /<font color=\"#666666\"><a href=\"(http:\/\/www\.zshare\.net\/[^\/]+\/[^\/]+)/;
    wft_uploader_username = /<small>Logged in as: ([^<]+)<\/small>/;
  })
  main = ({
    if (wft_delete_url ) {
        DB["web_file_transfer"]["wft_upload_id"] = wft_upload_id[0];
        DB["web_file_transfer"]["wft_delete"] = wft_delete_url[0]+"-"+wft_delete_url[1];

        DB["web_file_transfer"]["wft_site_name"] = "zshare.net";
        DB["web_file_transfer"]["transfer_type"] = "upload";

        if (wft_file_name) {
          DB["web_file_transfer"]["wft_filename"] = wft_file_name[0];
        }

        if (wft_url) {
          DB["web_file_transfer"]["wft_url"] = wft_url[0];
        }
        if (wft_uploader_username) {
          DB["web_file_transfer"]["uploader_username"] = wft_uploader_username[0];
        }
        DB.apply();
    } else {
       logger.debug("filetransfer/web/zshare.net/upload/response: Host regexs didn't match");
    }
    return true;
});
```

Meta-data Extracting Fingerprints

- All you do is tell XKS when to start extracting meta-data

```
appid('filetransfer/web/zshare_net/upload/response', 5.0)=
        http_title('zSHARE') and 'zshare.net/delete.html'
        : c++
```

Meta-data Extracting Fingerprints

- Use Regular Expressions to tell it what to extract:

```
extractors : {{
  wft_file_name = /The\sfile\s<strong><font\scolor=\"#333333\">([^<]{1,300})\s</;
  wft_delete_url = /zshare.net\/delete.html\?([0-9]+)-([0-9a-zA-Z]{32})\"/;
  wft_upload_id =  /<font color=\"#666666\"><a href=\"http:\/\/www\.zshare\.net\/[^\/]+\/([0-9]+)[0-9a-f]{8}/;
  wft_url = /<font color=\"#666666\"><a href=\"(http:\/\/www\.zshare\.net\/[^\/]+\/[^\/]+)/;
  wft_uploader_username = /<small>Logged in as: ([^<]+)<\/small>/;
}}
```

Meta-data Extracting Fingerprints

- Finally tell it which database tables you want to store the information:

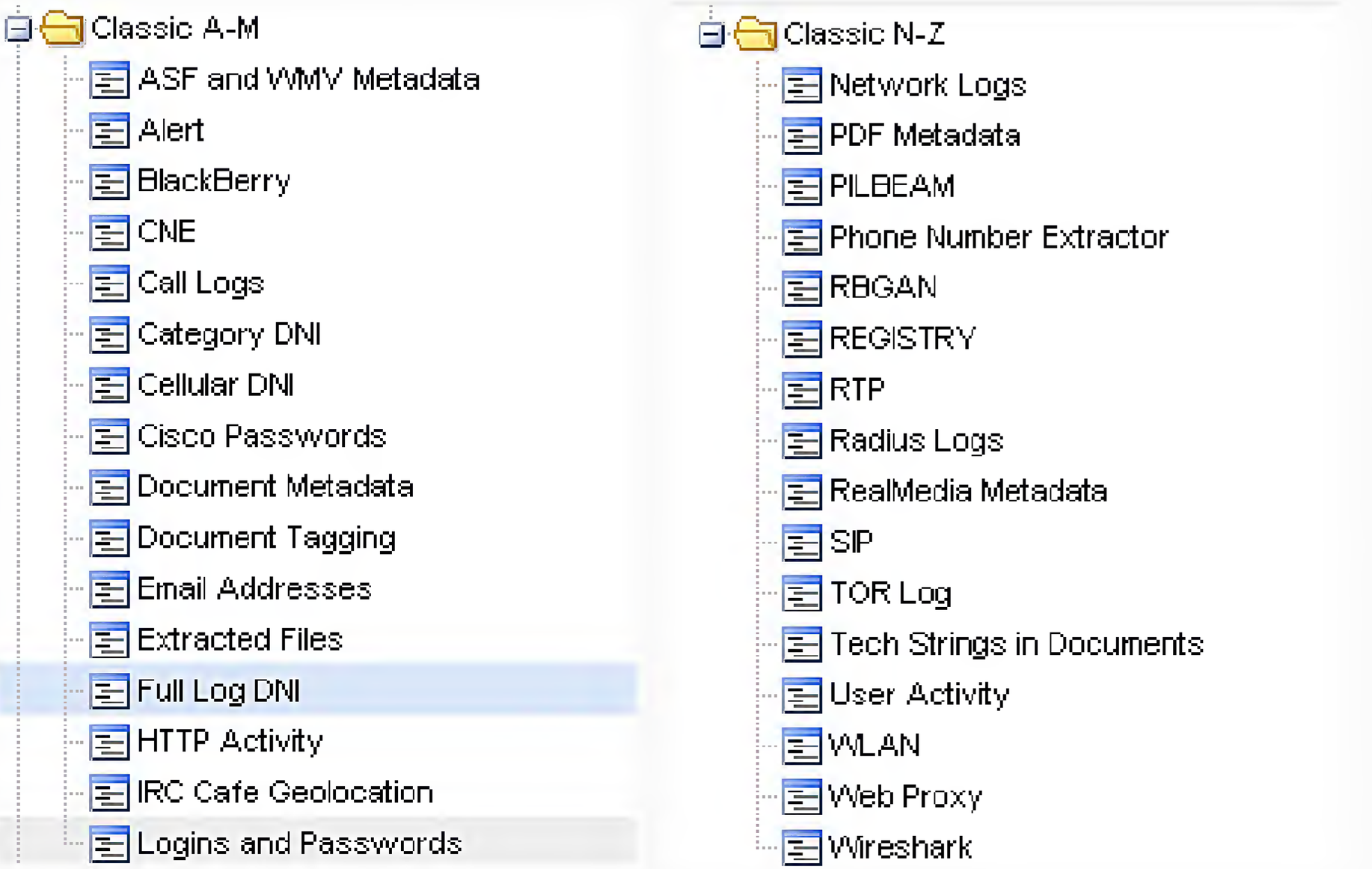
```
main = {{
  if (wft_delete_url ) {
      DB["web_file_transfer"]["wft_upload_id"] = wft_upload_id[0];
      DB["web_file_transfer"]["wft_delete"] = wft_delete_url[0]+"-"+wft_delete_url[1];

      DB["web_file_transfer"]["wft_site_name"] = "zshare.net";
      DB["web_file_transfer"]["transfer_type"] = "upload";
```

File URL	Filename
http://www.zshare.net/download/63719957f0b174c9f	**khi pics.zip**

Transfer Type	Upload ID	Delete ID	Site Name
upload	**63719957**	**7c8893b1bf04170771dca3e7f0756a26**	**zshare.net**

Meta-data Extracting Fingerprints

- What if the meta-data you want to extract doesn't fit nicely into any of the existing XKS meta-data tables?



Meta-data Extracting Fingerprints

- Define your own with the "Microplugin" query forms:

Meta-data Extracting Fingerprints

- Example MS2 KeyIDs

Search: Ms2 Extract Keyids

Query Name: [redacted]

Justification:

Recent Justifications

Additional Justification:

Miranda Number:

Datetime: 1 Day Start: 2010-05-03 00:00 Stop: 2010-05-04 23:59

ejKeyID:

Username<realm>:

IP Address: From [IP Address Field Builder]

IP Address: To [IP Address Field Builder]

Port: From

Port: To

Meta-data Extracting Fingerprints

Search: Ccne Byzantine Raptor Trojan3

Query Name: [redacted]

Justification:

Recent Justifications

Additional Justification:

Miranda Number:

Datetime: 1 Day Start: 2010-05-03 00:00 Stop: 2010-05-04 23:59

brt_decrypt:

brt_hostname:

brt_ipaddress:

brt_length:

brt_osversion:

brt_packet_type:

brt_sequence_num:

brt_username:

Username<realm>:

IP Address: From [IP Address Field Builder]

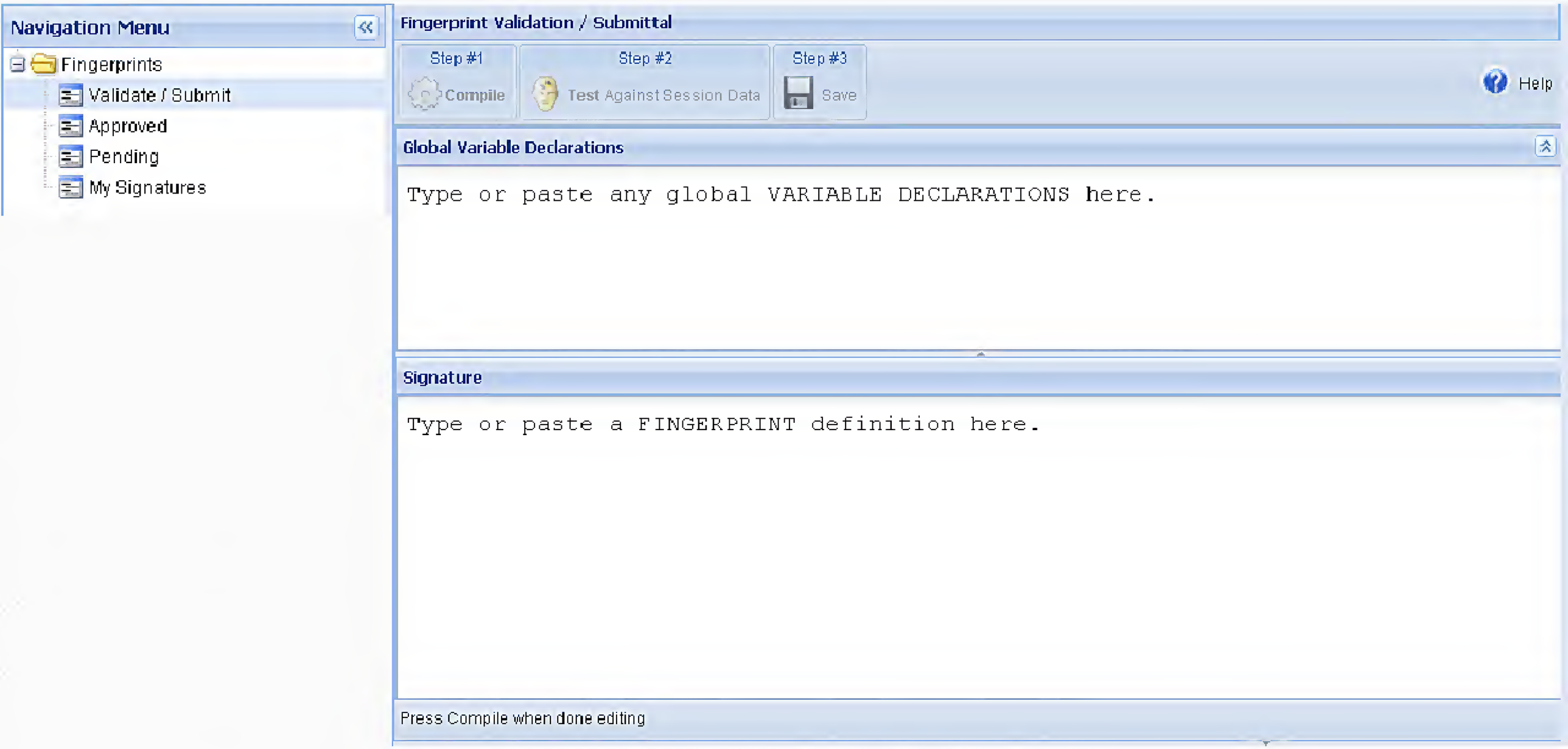
IP Address: To [IP Address Field Builder]

DB["

DB["

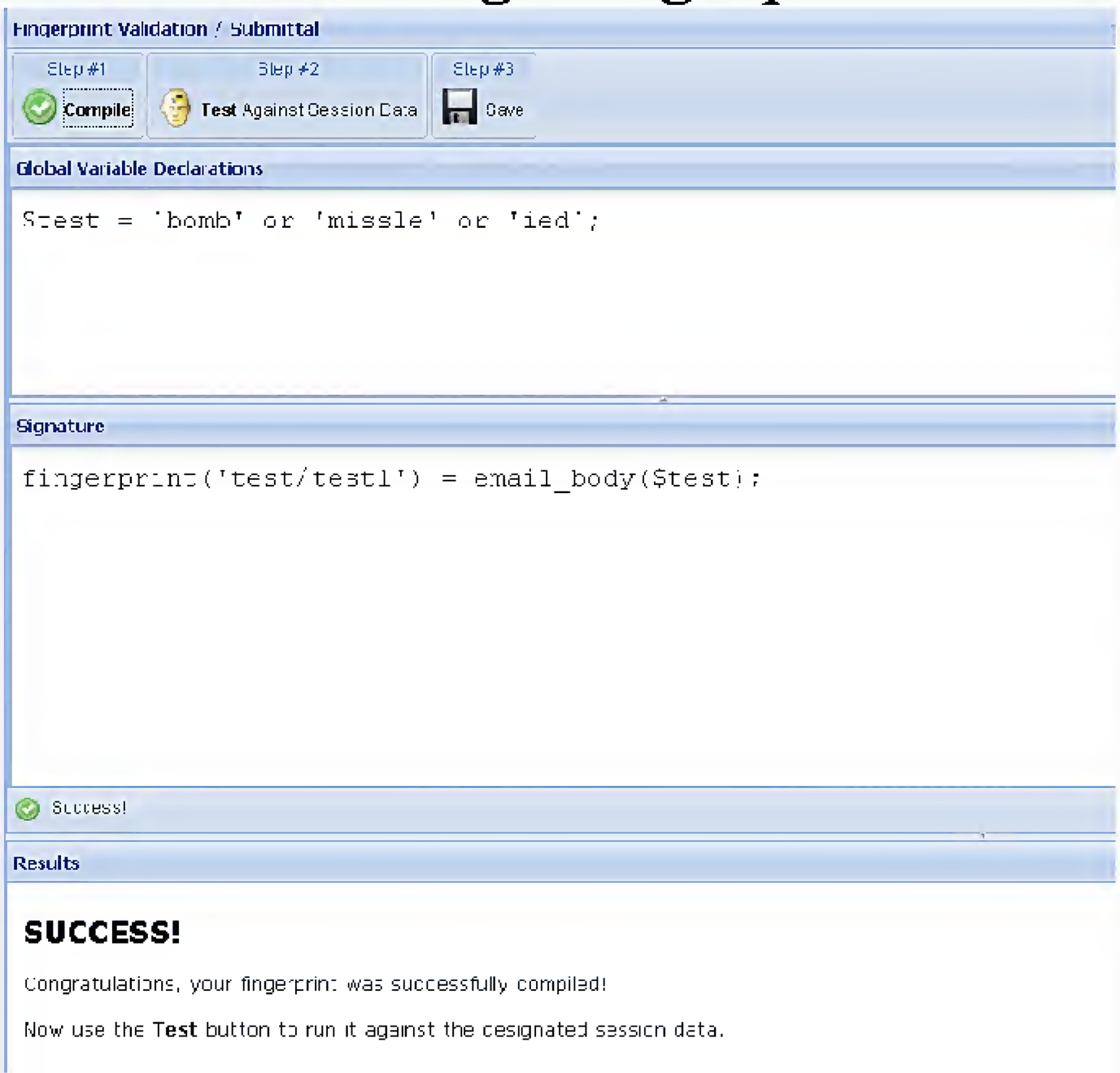
New Fingerprint GUI

- New XKS Fingerprint GUI allows analysts to directly test, submit and manage fingerprints through the web



New Fingerprint GUI

- New XKS Fingerprint GUI allows analysts to directly test, submit and manage fingerprints through the web



Questions?

Syntax Rules

- The definition of the fingerprint will look like this:

fingerprint('test/blah/something', owner = '[redacted]') =

Note the single quotes needed for the fingerprint name and owner

Syntax Rules

- Secondly every fingerprint definition must be completed by a semi-colon.

fingerprint('test/blah/something', owner = [redacted]) =
'badguy' ;

Syntax Rules

- Variables also must be completed by a semi-colon.

```
$badguy =
  'bomb' or 'gun' or 'weapon' ;
fingerprint('test/blah/something', owner = [redacted]) =
              $badguy;
```

Syntax Rules

- Definitions and Variables can span multiple lines

```
$badguy =
  'bomb' or
   'gun' or
   'weapon' ;
fingerprint('test/blah/something', owner = [redacted]) =
                $badguy;
```